



# UNITED STATES PATENT AND TRADEMARK OFFICE

MN

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/017,183	12/07/2001	Coach Wei	EMC-06-235(PRO)ORD1	5883
24227	7590	05/11/2007	EXAMINER	
EMC CORPORATION			ZHEN, LIB	
OFFICE OF THE GENERAL COUNSEL			ART UNIT	PAPER NUMBER
176 SOUTH STREET			2194	
HOPKINTON, MA 01748			MAIL DATE	DELIVERY MODE
			05/11/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/017,183	WEI, COACH	
	<b>Examiner</b>	<b>Art Unit</b>	
	Li B. Zhen	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### **Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

1)  Responsive to communication(s) filed on 21 February 2007.  
2a)  This action is **FINAL**.                            2b)  This action is non-final.  
3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

4)  Claim(s) 1-18 and 22-25 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5)  Claim(s) \_\_\_\_\_ is/are allowed.

6)  Claim(s) 1-18 and 22-25 is/are rejected.

7)  Claim(s) \_\_\_\_\_ is/are objected to.

8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

9)  The specification is objected to by the Examiner.

10)  The drawing(s) filed on \_\_\_\_\_ is/are: a)  accepted or b)  objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a)  All b)  Some \* c)  None of:  
1.  Certified copies of the priority documents have been received.  
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_.  
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)  Notice of References Cited (PTO-892)  
2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3)  Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.  
4)  Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.  
5)  Notice of Informal Patent Application  
6)  Other: \_\_\_\_\_.

### **DETAILED ACTION**

1. Claims 1 – 18 and 22 – 25 are pending in the current application.

#### ***Continued Examination Under 37 CFR 1.114***

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 02/21/2007 has been entered.

#### ***Claim Objections***

3. Claims 1 – 23 are objected to because of the following informalities: the abbreviations (e.g., GUI, API, AWT, HTTP, HTTPS, WAP, XML, PDA) in the claims should be defined. Appropriate correction is required.

#### ***Oath/Declaration***

4. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because: the oath or declaration was not signed by all of the actual inventors. Specification submitted on 12/07/2001 and application data sheet submitted on 06/30/2006 identifies Coach Wei and Zakir

Magdum as inventors. However, the oath or declaration submitted on 03/11/2002 is not signed by Zakir Magdum.

***Claim Rejections - 35 USC § 112***

5. Claims 2 – 6 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.
6. Claims 2 – 6 contains the trademark/trade name Java, Java Swing, Windows, X Windows, Windows CE and Palm. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe either a programming language or an operating system and, accordingly, the identification/description is indefinite.

***Response to Arguments***

7. In response to the Final Office Action dated 10/23/2006, applicant argues:

(1) Bahrs does not teach or suggest an application “developed once and deployed multiple times” and therefore does not have the advantages provided by Applicants’ claimed invention [p. 11, 13 and 15];

(2) JTC is a process, architectural pattern, and implementation guide to developers on how to build applications, and in particular, Internet style thin clients. Thus, Bahrs provides an architectural pattern that may be used to create applications [p. 12, 14 and 16];

(3) Bahrs presents a method “on how to build applications” where the current invention enables an application to be “developed once and deployed multiple times” [p. 12, 14 and 16]; and

(4) Instead of Bahrs method of using an architectural pattern to guide developers on how to build applications, the current invention enables the application to be developed once and deployed multiple times [p. 13, 15 and 16].

In response to argument (1), examiner respectfully disagrees and notes that Bahrs teaches a `ViewControllerImpl` that can be a Java bean and implements the `ViewController` and `JTC` interfaces [col. 19, lines 42 – 56]. A Java bean component can be developed once and then deployed on multiple platforms without recompilation or source code modification [p. 4, EJB Technology design goals of Nordby]. Therefore, when the components of Bahrs are implemented as Java beans, the applications of Bahrs can be developed once and deployed multiple times.

In response to argument (2), examiner notes that Bahrs teaches a `ViewControllerImpl` that implements the `ViewController` and `JTC` interfaces [col. 19, lines

42 – 56]. Although JTC is a process, architectural pattern, and implementation guide to developers on how to build application, the ViewControllerImpl is an implementation of the ViewController and JTC interfaces that allows applications to be developed once and deployed multiple times. In addition, Bahrs teaches that the architectural pattern can be used on a server [col. 37, lines 50 – 67] and the application containing the view controller is located on the server [col. 36, line 65 – col. 37, line 15]. Therefore, Bahrs teaches an architectural pattern and implementation a graphical user interface API [ViewControllerImpl] that is executing on the server.

As to argument (3), examiner disagrees and notes that Bahrs teaches an architectural pattern and implementation a graphical user interface API [ViewControllerImpl] that is executing on the server [col. 36, line 65 – col. 37, line 15], see also response to argument (2) above.

As to argument (4), examiner respectfully disagrees and notes that Bahrs teaches an architectural pattern and implementation a graphical user interface API [ViewControllerImpl] that is executing on the server [col. 36, line 65 – col. 37, line 15]. In addition, Bahrs teaches a ViewControllerImpl that can be a Java bean and implements the ViewController and JTC interfaces [col. 19, lines 42 – 56]. A Java bean component can be developed once and then deployed on multiple platforms without recompilation or source code modification [p. 4, EJB Technology design goals of Nordby]. Therefore, when the components of Bahrs are implemented as Java beans, the applications of Bahrs can be developed once and deployed multiple times.

***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. **Claims 1-8, 11, 13-18 and 22-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,901,554 to Bahrs et al. [hereinafter Bahrs, cited in the previous office action] in view of “What are Enterprise JavaBeans components?: Part 1: The history and goals of EJB architecture” [hereinafter Nordby].**

10. As to claim 1, Bahrs teaches the invention substantially as claimed including a method for delivering applications over a network in which the business logic of the application [business logic; col. 31, lines 5 – 15 and col. 14, lines 23 – 36] is running on the backend server [a server 104; col. 12, lines 16 – 43; server side business logic, col. 31, lines 5 - 15], the method comprising the steps of:

having the application invoke a GUI API to present the application's user interface [a client browser invokes a URL submit, the web server obtains the request and passed control to a servlet. The servlet obtains a key/value pair list of values entered in the HTML client. This list is passed to the ViewController alternate view being displayed; col. 38, lines 1 – 19];

replacing the GUI API with a re-implemented [replacement may be accomplished by creating the developer's own implementation of ViewControllerBaseImpl that implements the methods getComponent( ), setEnabled(boolean enable), and setVisible(boolean visible), col. 20, lines 33 – 52; overriding methods of the ViewController class, col. 31, line 53 – col. 32, line 22; examiner notes that when the methods of the ViewController class is overridden with the developer own implementation, the View Controller class is re-implemented] network aware GUI API [ViewController interface 3902 extends JTC interface 3904; col. 35, lines 45 – 54 and col. 44, line 13 – 50] running on a backend server [application containing the view controller may be located on the server; col. 36, line 65 – col. 37, line 15] which translates the application's presentation layer information [col. 47, line 63 – col. 48, line 15 and col. 36, line 65 – col. 37, line 16] into a pre-determined format based messages [Object data may take various forms, such as Extensible Markup Language (XML), String, Hypertext Markup Language (HTML), key/value, Remote Method Invocation (RMI), J/XFS, RS232; col. 17, lines 25 – 39] which describe a Graphical User Interface [col. 48, lines 40 – 60 and col. 53, lines 3 – 20], event processing registries [data is passed via different events, such as ViewEvent 510, RequestEvent 522, and RequestEvent 526; col. 17, lines 25 – 39] and other related information [object handling placement of components will register as a listener for notifications to place objects on the screen; col. 24, lines 36 – 59], the presentation layer of the application in a high level, object level messages [col. 16, line 57 – col. 17, line 15];

sending such messages to the client device via a network [col. 41, line 66 – col. 42, line 19; col. 48, lines 40 – 60 and col. 53, lines 3 – 20];

processing the messages and rendering a user interface by a client-side program [If the major code for the TopEvent is message, then the message is displayed for the application (step 8418); col. 49, lines 25 – 33], which delivers a user experience for that device according to the capability of the specific client device [mechanism for creating the HTML view is application dependent/screen dependent; col. 37, line 50 – 67];

rendering the user interface on the client device [ViewController 502 basically provides a reusable GUI element; col. 15, line 52 – col. 16, line 13];

transmitting a plurality of user input and client-side events back to the server by the client-side program [col. 36, lines 17 – 28] via a predetermined protocol [col. 14, lines 36 – 65];

processing the user input and client-side events on the backend server [col. 26, lines 1 – 20 and col. 16, line 56 – col. 17, line 15], translating such events and inputs as if they were locally generated [ViewEvents generated in the ViewControllers 12302 being handled by the ApplicationMediator 12304 and translated into appropriate RequestEvents; col. 65, lines 23 – 41], and sending such translated events and inputs to the application for processing [RequestEvents are passed on to the destination 12308 via the transported 12306; col. 65, lines 23 – 41];

encoding and routing the output of the application to the client device using the predetermined messaging format [col. 16, line 57 – col. 17, line 15]; and

further processing the output by the client-side program to refresh the Graphical User Interface thereat [the return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user; col. 16, line 57 – col. 17, line 15]. Bahrs discloses that the ViewControllerImpl that implements the ViewController and JTC interfaces is usually a Java Component or Container or bean [col. 19, lines 42 – 56]. Bahrs does not specifically disclose applications that are developed once and deployed multiple times.

However, Nordby teaches an EJB component can be developed once and then deployed on multiple platforms without recompilation or source code modification [p. 4, EJB Technology design goals].

Bahrs teaches that the ViewControllerImpl that implements the ViewController and JTC interfaces is usually a Java Component or Container or bean. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to implement the ViewControllerImpl of Bahrs as a Java bean and provide applications that can be developed once and deployed multiple times because this simplifies development of middleware components that are transactional, scalable, and portable [p. 1, 4<sup>th</sup> paragraph of Nordby] and provides a robust, scalable environment that can support mission-critical enterprise information systems [p. 1, 5<sup>th</sup> paragraph of Nordby].

11. As to claim 22, Bahrs as modified teaches a system for distributing an application [col. 14, lines 23 – 36 of Bahrs] including at least a server [a server 104; col. 12, lines 15 – 45 of Bahrs], at least a client device [clients 108, 110, and 112; col. 12, lines 16 – 43

of Bahrs], and a communication means [network 102; col. 12, lines 16 – 45 of Bahrs], the system comprising:

    a presentation layer of the application [ViewController; col. 15, line 52 – col. 16, line 12 of Bahrs] written using a server-side API [col. 19, lines 12 – 30 of Bahrs] based network programming model [col. 28, lines 42 – 67 of Bahrs];

    a business logic layer of the application [business logic; col. 31, lines 5 – 15 and col. 14, lines 23 – 36 of Bahrs] and a data layer of the application [data model; col. 35, line 57 – col. 36, line 6 of Bahrs] both of which are written with the server-side API and running on the server [a server 104; col. 12, lines 16 – 43; server side business logic, col. 31, lines 5 - 15 of Bahrs]; and where

        the server-side API having a supporting infrastructure that sends [Object data may take various forms, such as Extensible Markup Language (XML), String, Hypertext Markup Language (HTML), key/value, Remote Method Invocation (RMI), J/XFS, RS232; col. 17, lines 25 – 39 of Bahrs] the application's user interface information [col. 47, line 63 – col. 48, line 15 of Bahrs] to a client device for presentation [col. 48, lines 40 – 60 and col. 53, lines 3 – 20 of Bahrs], handles communications problems [col. 43, lines 15 – 36 of Bahrs], renders the application's user interface [ViewController 502 basically provides a reusable GUI element; col. 15, line 52 – col. 16, line 13 of Bahrs] and dispatches necessary user input events back to the server for processing [col. 18, line 63 – col. 19, line 13 of Bahrs];

        wherein use of the system enable the application [col. 19, lines 42 – 56 of Bahrs] to be developed once and deployed multiple times [EJB component can be developed

once and then deployed on multiple platforms without recompilation or source code modification; p. 4, EJB Technology design goals of Nordby].

12. As to claim 23, Bahrs as modified teaches an apparatus for distributing an application over a network [col. 14, lines 23 – 36 of Bahrs] where the apparatus includes:

a server [a server 104; col. 12, lines 15 – 45 of Bahrs];  
a client device [clients 108, 110, and 112; col. 12, lines 16 – 43 of Bahrs];  
a network communication means [network 102; col. 12, lines 16 – 45 of Bahrs];  
a re-implemented [replacement may be accomplished by creating the developer's own implementation of ViewControllerBaselImpl that implements the methods getComponent( ), setEnabled(boolean enable), and setVisible(boolean visible), col. 20, lines 33 – 52 of Bahrs; overriding methods of the ViewController class, col. 31, line 53 – col. 32, line 22 of Bahrs; examiner notes that when the methods of the ViewController class is overridden with the developer own implementation, the View Controller class is re-implemented] network based API module that is used to transparently replace the API on which the application was developed [ViewController interface 3902 extends JTC interface 3904; col. 35, lines 45 – 54 and col. 44, line 13 – 50 of Bahrs];  
a first means for running an application of the plurality of applications where a business logic [business logic; col. 31, lines 5 – 15 and col. 14, lines 23 – 36 of Bahrs] of the application runs on the server [a server 104; col. 12, lines 16 – 43; server side business logic, col. 31, lines 5 - 15 of Bahrs];

a second means for replacing the API [col. 20, lines 33 – 52 of Bahrs; overriding methods of the ViewController class, col. 31, line 53 – col. 32, line 22 of Bahrs] of each of the plurality of applications with the network based API [Interfaces extending JTC are ViewController, ApplicationMediator, and Destination; col. 44, lines 13 – 51 of Bahrs] so that each of the applications' logic runs on the server [application containing the view controller may be located on the server; col. 36, line 65 – col. 37, line 15 of Bahrs];

a third means for using the network based API to create a display for an application on the client device [ViewController 502 basically provides a reusable GUI element; col. 15, line 52 – col. 16, line 13 of Bahrs];

a fourth means for transferring the user interactions on the client device to the server [col. 18, line 63 – col. 19, line 13 of Bahrs], calculating the appropriate response to the input [deliver the information to the server's service for processing; col. 16, line 56 – col. 17, line 15 of Bahrs], and transmitting the appropriate response to the client machine [response data will be returned to the Transporter 524 in a RequestEvent; col. 16, line 56 – col. 17, line 15 of Bahrs];

a fifth means for updating the display of the application on the client device based on the responses from the server [return data may be sent to ViewController 502 to refresh the view displayed on the screen to the user; col. 16, line 56 – col. 17, line 15 of Bahrs];

wherein use of the re-implemented network aware API enables the application [col. 19, lines 42 – 56 of Bahrs] to be developed once and deployed multiple times [EJB component can be developed once and then deployed on multiple platforms without

recompilation or source code modification; p. 4, EJB Technology design goals of Nordby].

13. As to claim 2, Bahrs teaches the GUI API and the event processing API are Java Foundation Classes [col. 14, lines 36 – 65].

14. As to claim 3, Bahrs teaches the client-side program is a computer program based on Operating System's API, such as Windows API, or X Windows API [col. 34, lines 30 – 39 and col. 13, lines 43 – 60].

15. As to claim 4, Bahrs teaches the client-side program is a wireless device program written using the device's Operating System's API, such as Palm API and Windows CE API [col. 15, lines 26 – 52 and col. 14, lines 1 – 17].

16. As to claim 5, Bahrs teaches the client-side program is Java program written using Java API [col. 14, lines 36 – 65 and col. 15, lines 25 – 52].

17. As to claim 6, Bahrs teaches the JAVA API is AWT, Personal Java, Java 2 Micro Edition based GUI API or Java Swing [col. 14, lines 36 – 65 and col. 35, lines 45 – 54 and col. 44, line 13 – 50].

18. As to claim 7, Bahrs teaches the predetermined protocol is HTTP [JTC has natural support for multiple protocols, such as, for example IIOP, RMI, Sockets, HTTP, HTTPS, and Files; col. 15, lines 26 – 52].

19. As to claim 8, Bahrs teaches the predetermined protocol is HTTPS [JTC has natural support for multiple protocols, such as, for example IIOP, RMI, Sockets, HTTP, HTTPS, and Files; col. 15, lines 26 – 52].

20. As to claim 11, Bahrs teaches the predetermined messaging format is based on XML [col. 17, lines 25 – 38 and col. 37, line 50 – 67].

21. As to claim 13, Bahrs teaches the network is the Internet [col. 12, lines 16 – 43].

22. As to claim 14, Bahrs teaches the network is a local area network [col. 12, lines 16 – 43].

23. As to claim 15, Bahrs teaches the local area network is a bandwidth-limited slow speed network [col. 1, line 58 – col. 2, line 15].

24. As to claim 16, Bahrs teaches the network includes a wireless network [col. 15, lines 25 – 52].

25. As to claim 17, Bahrs teaches the client device is selected from the group consisting of workstations, desktops, laptops, PDAs, wireless devices and other edge devices [col. 15, lines 25 – 52].
26. As to claim 18, Bahrs teaches the server and the client device are combined into one entity [col. 17, lines 61 – 67 and col. 31, lines 5 – 15].
27. As to claim 24, Bahrs teaches the application code is not modified when distributing the application [col. 14, lines 23 – 36] and the application code is not distributed to the client device [business logic and central data management of an application should be separated out from the JTC application; col. 31, lines 5 – 15].
28. As to claim 25, Bahrs teaches distributing a plurality of pre-existing applications [col. 14, lines 23 – 36].
29. **Claims 9, 10 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bahrs and Nordby further in view of U.S. Patent No. 6,615,131 to Rennard et al. [hereinafter Rennard, cited in the previous office action].**
30. As to claim 9, Bahrs as modified teaches wireless devices [col. 15, lines 26 – 52 of Bahrs] and multiple protocols [col. 15, lines 26 – 52 of Bahrs] but does not specifically disclose the WAP protocol.

However, Rennard teaches Java user interfaces [col. 8, line 64 - col. 9, line 37] and the WAP protocol [Wireless Application Protocol; col. 7, line 64 – col. 8, line 13].

It would have been obvious to a person of ordinary skill in the art at the time of the invention to apply the teaching of the WAP protocol to the invention of Bahrs because the Wireless Markup Language (WML) in the Wireless Application Protocol includes navigation and event-handling models that allow an author to specify the processing of user agent events [col. 7, line 63 – col. 8, line 23 of Rennard]. In addition, the WAP protocol allows the creation of a WML foundation class that reduces the amount of code that must be written to create a WML deck [col. 8, lines 21 – 39 of Rennard].

31. As to claim 10, Bahrs as modified teaches the predetermined protocol is proprietary [col. 6, line 57 – col. 7, line 2 of Rennard].

32. As to claim 12, Bahrs as modified teaches the predetermined messaging format is proprietary [col. 6, line 57 – col. 7, line 2 of Rennard].

#### **CONTACT INFORMATION**

33. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Li B. Zhen  
Examiner  
Art Unit 2194

LBZ



5/4/2007